DOI: 10.18372/2410-7840.20.13074
УДК 004.8.565.5

МЕТОД РОЗРОБКИ АРХІТЕКТУРИ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ, ПРИЗНАЧЕНОЇ ДЛЯ РОЗПІЗНАВАННЯ КОМП'ЮТЕРНИХ ВІРУСІВ

Ігор Терейковський, Олег Заріцький, Людмила Терейковська, Володимир Погорелов

Стаття присвячена вирішенню задачі вдосконалення систем розпізнавання комп'ютерних вірусів. Хоча використовуються системи антивірусного захисту вже не одне десятиліття, їх розробкою займається багато висококваліфікованих фахівців, а створенню відповідної науково-методичної бази присвячена велика кількість робіт, проте практичний досвід і відомі випадки успішних вірусних атак на вітчизняні та закордонні комп'ютерні системи та мережі вказують на наявність в сучасних антивірусах розпізнавання серйозних недоліків. Показано, що виправити ряд недоліків можливо шляхом удосконалення математичного забезпечення процедури розпізнавання за рахунок застосування сучасних нейромережевих моделей на базі глибоких нейронних мереж. Запропоновано метод розробки архітектури глибокої нейронної мережі, призначеної для розпізнавання вірусів. На відміну від існуючих метод дозволяє уникнути в процесі розробки нейромережевої моделі довготривалих чисельних експериментів, спрямованих на визначення доцільності її застосування та на оптимізацію її структурних параметрів. Шляхом чисельних експериментів з використанням опублікованої компанією Microsoft бази даних комп'ютерних вірусів BIG-2015 показано, що метод дозволяє побудувати нейромережеву модель, яка забезпечує похибку розпізнавання, співрозмірну з похибкою сучасних систем розпізнавання комп'ютерних вірусів. Визначено, що перспективи подальших досліджень пов'язані з адаптацією запропонованого методу до застосування глибоких нейронних мереж в поведінкових аналізаторах.

Ключові слова: *захист інформації, комп'ютерний вірус, нейромережева модель, глибока нейронна мережа, розріджений автокодувальник..*

Вступ

В сучасних умовах системи антивірусного захисту є одним з основних засобів захисту інформації більшості комп'ютерних систем і мереж [1, 5]. Хоча використовуються такі системи вже не одне десятиліття, їх розробкою займається багато висококваліфікованих фахівців, а створенню відповідної науково-методичної бази присвячена велика кількість робіт, проте практичний досвід і дані [3] вказують на наявність в сучасних антивірусах розпізнавання серйозних недоліків. Основним з них є недостатня точність розпізнавання всієї номенклатури комп'ютерних вірусів. На наявність цієї вади вказують як результати [2, 6-11], так і відомі випадки успішних вірусних кібератак на вітчизняні та закордонні комп'ютерні системи і мережі.

При цьому важливим напрямком підвищення точності розпізнавання є "інтелектуалізація" методів розпізнавання за рахунок використання теорії штучних нейронних мереж (НМ) [1, 5]. Перспективність вказаного напрямку підтверджується окремими вдалими застосуваннями НМ в засобах розпізнавання комп'ютерних вірусів (антивірус з відкритим програмним кодом ClamAV, стартап Deep Instinct) та великою кількістю відповідних теоретико-практичних робіт, огляд яких наведено в [1, 5, 7]. Разом з тим, недостатня точність розпізнавання та недостатня адаптованість до умов експлуатації, закритість використаних рішень значно обмежують сферу їх застосування. При цьому постійний прогрес в області теорії нейронних мереж вказує на можливість значного вдосконалення апробованих засобів розпізнавання. Цим пояснюється актуальність досліджень в області вдосконалення існуючих нейромережевих засобів, що за рахунок використання сучасних теоретичних рішень дозволили б забезпечити ефективне розпізнавання комп'ютерних вірусів.

Аналіз існуючих досліджень. Постановка задачі

Аналіз науково-практичних робіт, присвячених вдосконаленню систем розпізнавання комп'ютерних вірусів, дозволяє стверджувати, що в таких системах НМ застосовуються для виявлення комп'ютерних вірусів на основі узагальнення статистичних даних, відображених в навчальних прикладах [2, 9-11]. При цьому розпізнавання комп'ютерних вірусів за допомогою НМ в основному зводиться до оцінок величин множини контрольованих параметрів. Якщо виставлена за допомогою НМ оцінка знаходиться в певному діапазоні, то вважається, що комп'ютерний вірус розпізнано. В протилежному випадку вважається, що в комп'ютерній системі віруси відсутні. Перелік контрольованих параметрів залежить від типу системи розпізнавання. Для поведінкового аналізатора перелік може визначатись набором ознак викликів потенційно небезпечних функцій приклад-

ного програмного інтерфейсу операційної системи. Для антивірусного сканера перелік ознак може співвідноситься із сигнатурами комп'ютерних вірусів. Також можна сформувати висновок про те, що більшість відповідних науково-практичних робіт присвячені або застосуванню перспективних нейромережевих моделей, або проблемі адаптації параметрів нейромережевих засобів розпізнавання вірусів до очікуваних умов застосування. Слід зазначити, що для поглиблення результатів аналізу також були розглянути науково-практичні роботи, присвячені розробці нейромережевих засобів оцінки параметрів безпеки інформаційних систем. Так, в роботі [5] сформовано базову множину критеріїв ефективності виду нейромережевої моделі, що використовується для оцінки параметрів безпеки. Визначені шляхи розширення цієї множини. Також в роботі створено методологію розробки нейромережевих засобів для оцінки параметрів безпеки ресурсів інформаційних систем, що може бути використана при побудові систем розпізнавання шкідливого програмного забезпечення. Крім того в роботі [5] розроблено метод оцінки ефективності нейромережевих засобів розпізнавання Інтернет-орієнтованих кібератак, до складу яких також можна віднести і Інтернет-орієнтовані комп'ютерні віруси.

Метод визначення фрагментів програмного коду описаний в роботі [3]. Метод застосовується для визначення переліку та оцінки значень вхідних параметрів НМ, що використовуються в системах детектування шкідливого програмного забезпечення. Також в роботі [3] наведено опис та результати експериментів по розпізнаванню шкідливого програмного забезпечення. Для розпізнавання використано НМ типу двохшарового персептрону (ДШП). Аналіз наведених результатів підтверджує перспективність запропонованого методу. Можна зробити висновок про використання в методі процедури попередньої обробки вхідних параметрів НМ, яка підвищує їх інформативність.

Модель топографічної карти Кохонена для розпізнавання комп'ютерних вірусів (МТК), розроблена в роботі [1]. Модель призначена для використання в антивірусних сканерах. Передбачено блок попередньої обробки вхідних параметрів. Вибір типу моделі реалізовано шляхом порівняльних числових експериментів. В якості критерію порівняння використано термін навчання. Оптимізація параметрів та процедури навчання нейромережевої моделі не проводилась.

В роботі [10] запропоновано систему розпізнавання комп'ютерних вірусів на основі нейромережевого аналізу нормалізованих сигнатур. Декларується точність розпізнавання в межах 80-91%. Вказується на можливість розпізнавання поліморфних вірусів. В якості базової нейромережевої моделі використано ДШП. Схожі результати отримані і в роботах [2, 3, 11] де для розпізнавання також використано ДШП з одним або двома вихідними нейронами. При цьому вхідні нейрони співвідносяться із параметрами, що характеризують структуру PE-файлів. Основною відмінністю між результатами [2, 3, 10, 11] є використання різних підходів до попередньої обробки вхідних параметрів НМ. Також в роботі [2] описані експерименти, що свідчать про точність розпізнавання комп'ютерних вірусів, сигнатури яких представлені в базі даних malwr на рівні 91%. Відзначимо, що в роботах [2, 3, 10, 11] механізму оптимізації структури багатошарового персептрону та механізму формування навчальної вибірки не наведено.

В роботі [4] задекларовано створення нейромережевого методу автоматичної генерації сигнатур комп'ютерних вірусів. В методі використовується глибока нейронна мережа (ГНМ) на основі автокодера. Мережа складається із 8 шарів, кожен із яких містить 30 нейронів. Метод не передбачає етапу навчання НМ на маркерованих навчальних даних, хоча цей етап вважається обов'язковим з точки зору забезпечення високої точності розпізнавання. При цьому декларується точність розпізнавання 98%.

В роботі [9] розроблено підхід до розпізнавання шкідливого програмного забезпечення за допомогою згорткових нейронних мереж (ЗНМ), що є одним із різновидів ГНМ, пристосованим для аналізу графічної інформації. Підхід передбачає подання піддослідного програмного забезпечення у вигляді сірого масштабованого зображення. Проведені експерименти, в яких досліджувався вплив структурних параметрів ЗНМ на точність розпізнавання. При цьому у всіх випадках розмір вхідного нейронного шару залишався 32x32. Вказано, що досягнута точність розпізнавання 93.86%. Зазначимо, що експериментально досліджено тільки 3 варіанти структурних рішень ЗНМ. Також недостатньо теоретично обґрунтована доцільність представлення коду піддослідного програмного забезпечення у вигляді двовимірного графічного зображення.

В роботі [7] сформовано підхід до визначення вхідних параметрів НМ, призначеної для розпізнавання комп'ютерних вірусів. Підхід передбачає встановлення аналогії між вхідними нейронами та

викликами потенційно небезпечних API-функцій операційної системи. Під потенційно небезпечними розуміють API-функції без використання яких неможливо завдати шкоду комп'ютерній системі або/та забезпечити саморозповсюдження вірусу.

В роботі [8] розроблено нейромережеву модель, призначену для розпізнавання мережевих кібератак на ресурси комп'ютерних систем та мереж. Показано доцільність використання ГНМ. Це пояснюється тим, що даному виду нейромережевої моделі притаманна висока здатність до навчання, високі обчислювальні можливості та висока адаптованість до особливостей умов застосування. Наведено результати експериментів розпізнавання мережевих кібератак, сигнатури яких представлені в базі даних NSL-KDD.

В результаті проведеного аналізу можна стверджувати, що одним із найбільш перспективних напрямків підвищення ефективності антивірусних систем є застосування в них нейромережевих засобів розпізнавання на базі ГНМ. При цьому в доступній літературі відсутнє теоретичне обґрунтування доцільності такого використання ГНМ. Також не наведене обґрунтування адаптації архітектурних параметрів ГНМ до умов задачі розпізнавання комп'ютерних вірусів, що значно знижує ефективність відповідних нейромережевих засобів.

Метою даної роботи є забезпечення ефективності систем розпізнавання комп'ютерних вірусів на основі методу розробки архітектури глибокої нейронної мережі, адаптованої до умов застосування в антивірусних засобах.

1.1. Основна частина дослідження

Відповідно до сучасної методології побудови нейромережевих систем захисту інформації [5] прийнято, що для адаптації ГНМ до умов застосування в антивірусних засобах необхідно вирішити такі завдання: встановити доцільність використання ГНМ, розробити критерії визначення ефективності її архітектури, визначити найбільш ефективну архітектуру ГНМ, визначити архітектурні параметри і провести експериментальні дослідження спрямовані на верифікацію отриманих рішень.

Встановлення доцільності використання ГНМ. Як показують результати [5], основним фактором, який впливає на формування множини допустимих видів нейромережевих моделей, є забезпечення їх ефективного навчання. Для цього необхідно за допустимий час виконати наступні процедури: визначити множину вхідних і вихідних параметрів нейромережевої моделі, провести кодування зазначених вхідних та вихідних параметрів, створити навчальну вибірку, реалізувати процес навчання. Перша і друга процедури реалізується на підготовчих етапах розробки системи розпізнавання, тому їх вплив на формування множини допустимих архітектур ГНМ не розглядається. Основна увага акцентована на виконанні третьої та четвертої процедури.

На підставі даних [5, 8] можна зробити висновок про те, що основним обмеженням реалізації зазначених процедур є термін створення навчальної вибірки і навчання нейромережевої моделі. Також можна зробити висновок, що прийнятний термін такого створення визначається на підставі вимог до створення системи розпізнавання вірусів. Отже,

$$t_{\Sigma} \le t_d, \qquad (1)$$

де t_{Σ} – загальний термін навчання ГНМ, t_d – прийнятний термін створення нейромережевої системи розпізнавання.

Таким чином, допустимість використання *i*-ої архітектури ГНМ для розпізнавання вірусів можна задати за допомогою такого правила:

$$\text{If } t_{\Sigma}(\text{net}_i) \le t_d \rightarrow \text{net}_i \in \boldsymbol{Net_d}, \qquad (2)$$

де net_i – і-а архітектура глибокої нейронної мережі, $\boldsymbol{Net_d}$ – множина допустимих архітектур.

У першому наближенні можна прийняти до розгляду тільки ГНМ на базі багатошарового персептрону. Деталізувавши вираз (1), отримаємо:

$$t_{\Sigma}(\text{net}_i) = t_v + t_l(\text{net}_i), \qquad (3)$$

де t_v – час створення навчальної вибірки, $t_l(\text{net}_i)$ – час визначення параметрів моделі для і-ої архітектури.

Відзначимо, що в першому наближенні значення $t_l(\text{net}_i)$ приблизно дорівнює часу визначення вагових коефіцієнтів синаптичних зв'язків НМ. При цьому, з точки зору вирішення завдання визначення принципової можливості використання певної нейромережевої архітектури, створення навчальної вибірки зводиться до формування такої кількості навчальних прикладів, яке вважається достатнім для якісного навчання мережі.

Відповідно до [5], ця кількість залежить від кількості вхідних параметрів нейромережевої моделі і в базовому випадку розраховується так:

$$P_{min} \approx 10N_x, \qquad (4)$$

де P_{min} – мінімально допустима кількість навчальних прикладів, N_x – кількість вхідних параметрів нейромережевої моделі.

Також можна прийняти, що

$$t_v = \bar{t}_v P_{min}\ , \qquad (5)$$

де $\bar{t}_v$ – середній час створення одного навчального прикладу.

Очевидно, що величина $\bar{t}_v$ є індивідуальною для конкретного вірусу і залежить від багатьох факторів: організації процесу створення навчальної вибірки, апаратно-програмного забезпечення і т. ін. Визначити величину $\bar{t}_v$ можливо шляхом експертного оцінювання. Після підстановки (4) в (5) отримаємо:

$$t_v = 10\bar{t}_v N_x\ . \qquad (6)$$

У свою чергу, для приблизної оцінки часу визначення значень вагових коефіцієнтів синаптичних зв'язків нейромережевої моделі необхідно врахувати архітектуру цієї моделі, швидкість її апаратно-програмної реалізації, кількість навчальних прикладів, кількість вхідних і вихідних параметрів, а також допустиму величину помилки навчання.

Для *i*-го виду нейромережевої архітектури тривалість процесу визначення вагових коефіцієнтів можна оцінити так:

$$t_l(\text{net}_i) = \tau \times L_i \times W_i \times K_{o,i}\ , \qquad (7)$$

де τ – тривалість однієї навчальної ітерації для одного синаптичного зв'язку; W_i – кількість синаптичних зв'язків для *i*-го виду нейромережевої архітектури; L_i – кількість нейронів; $K_{o,i}$ – кількість навчальних ітерацій.

В роботі [1, 2] для оцінки тривалості навчання множини нейромережевих архітектур net, яке складається з нейромережевих моделей, що базуються на багатошаровому персептроні, запропоновано використовувати вираз

$$t_l(\mathbf{net}) \approx k\tau\, e^{-\chi\varepsilon} P^2 \left(N_x + N_y\right)^2, \qquad (8)$$

де $t_l(\mathbf{net})$ – тривалість визначення вагових коефіцієнтів, k – коефіцієнт пропорційності, χ – емпіричний коефіцієнт, ε – допустима помилка навчання.

Відзначимо, що вираз (8) отримано за умови послідовного розрахунку сигналів штучних нейронів, які входять до складу нейромережевої моделі, що характерно при її загальноприйнятій реалізації. Крім цього, прийнята передумова, що структура нейромережевої моделі та обчислювальні можливості нейромережевої архітектури достатні для отримання допустимої помилки навчання.

При заданій програмній реалізації нейромережевої моделі тривалість однієї обчислювальної операції процесу навчання в основному залежить від обчислювальної потужності апаратного забезпечення контуру розпізнавання вірусів в системі захисту інформаційних ресурсів.

Допустиму похибку навчання глибокої НМ можна розрахувати на підставі вимог до точності розпізнавання вірусів. У першому наближенні величини k і τ можливо визначити шляхом експертного оцінювання.

При визначенні принципової можливості використання нейромережевої моделі доцільно орієнтуватися на мінімально допустиму кількість навчальних прикладів. З огляду на вираз (8) і залежність (4), отримаємо:

$$t_l(\mathbf{net}) \approx 100k\tau\, e^{-\chi\varepsilon} N_x^2 \left(N_x + N_y\right)^2. \qquad (9)$$

Підставивши вирази (6, 9) у вираз (3) з урахуванням, що $k \approx 0{,}001$, $\chi \approx 1$, $\varepsilon \approx 0{,}05$, після тривіальних спрощень отримаємо:

$$t_\Sigma(\mathbf{net}) \approx 10N_x\left(\bar{t}_v + 0{,}01\tau N_x\left(N_x + N_y\right)\right), \qquad (10)$$

де $t_\Sigma(\mathbf{net})$ – загальний час навчання ГНМ.

Результати [3] вказують на те, що при розпізнаванні вірусів множина вхідних параметрів нейромережевої моделі не перевищує 300.

Оскільки на вхід ГНМ, крім безпосередньо зареєстрованих параметрів, можуть подаватися й інші параметри, в першому наближенні приймемо, що кількість вхідних параметрів нейромережевої моделі $N_x = 300..400$. При цьому кількість вихідних параметрів не перевищує 100. Ці передумови дозволяють модифікувати (10) наступним чином:

$$t_\Sigma(\mathbf{net}) \approx 4000\left(\bar{t}_v + 2100\tau\right). \qquad (11)$$

З урахуванням отриманого виразу (11) правило (2) можна деталізувати так:

$$\text{If } 4000\left(\bar{t}_v + 2100\tau\right) \le t_d \rightarrow \mathbf{net}_i \in \mathbf{Net}_d\ . \qquad (12)$$

Умова (12) визначає допустимість використання ГНМ для розпізнавання комп'ютерних вірусів.

Розробка критеріїв ефективності. За аналогією з [7, 10], будемо вважати, що серед множини допустимих базових архітектур *i*-а архітектура ГНМ є найбільш ефективною, якщо для неї функція ефективності прийме максимальне значення:

$$\max_{V_i} = \left\{ V_1, V_2, ..., V_I \right\}, \qquad (13)$$

де I – кількість архітектур ГНМ; V_i – функція ефективності *i*-ої архітектури ГНМ.

Розрахунок функції ефективності виконується так:

$$V_i = \sum_{k=1}^{K} \alpha_k R_k(net_i), \quad net_i \in \mathbf{Net}_d, \qquad (14)$$

де $\alpha_k = [0..1]$ – ваговий коефіцієнт *k*-го критерію ефективності, net_i – *i*-та архітектура ГНМ, K – кількість критеріїв ефективності, $R_k(net_i)$ – значення *k*-го критерію для ГНМ з *i*-ою архітектурою.

Відповідно до результатів [5, 8], під *k*-им критерієм визначення найбільш ефективної архітектури ГНМ будемо розуміти міру забезпечення в цій архітектурі *k*-ої вимоги задачі розпізнавання вірусів. З урахуванням [1], визначено, що вимоги до ГНМ характеризують їх здатність до навчання, обчислювальні можливості та особливості технічної реалізації. У свою чергу вимоги до навчання визначаються можливістю:

– Використовувати приклади з різною кількістю вхідних параметрів. Ця вимога істотно спрощує організацію процесу збору та попередньої обробки реальних статистичних даних.

– Використовувати навчальну вибірку, обсяг якої буде меншим за кількість вхідних параметрів, тобто на вибірці в ≤100 прикладів. Виконання цієї вимоги забезпечує можливість розпізнавати нові види вірусів, статистика по яких є обмеженою.

– Не пропорційного представлення в навчальній вибірці класів, що розпізнаються.

– Застосовувати навчальні приклади, в яких відсутній очікуваний вихідний сигнал.

– Ефективно навчатись на зашумлених навчальних прикладах.

– Засвоювати навченою ГНМ нових навчальних прикладів без повного перенавчання.

– Паралельного навчання. В цьому випадку ГНМ може навчатися по частинах.

– Стабільного навчання. В цьому випадку глибока нейронна мережа за прийнятний період навчання гарантовано забезпечує достатню помилку навчання.

– Мінімізації терміну навчання, який в основному визначається кількістю навчальних ітерацій.

– Забезпечити високий рівень автоматизації навчання, що при використанні якісної навчальної вибірки залежить від кількості емпіричних параметрів глибокої нейронної мережі.

– Можливістю подачі в ГНМ явних експертних знань.

Вимоги до обчислювальних («інтелектуальних») можливостей ГНМ визначаються:

– Відношенням кількості навчальних прикладів, які може запам'ятати НМ, до кількості синаптичних зв'язків в цій моделі. Зазначена вимога отримала назву обчислювальної потужності.

– Помилкою інтерполяції даних, що характеризує можливість правильного розпізнавання прикладів, які, хоча і не увійшли в навчальну вибірку, але значення параметрів яких знаходяться в межах значень параметрів навчальних прикладів.

– Помилкою екстраполяції даних, яка характеризує можливість правильного розпізнавання прикладів, значення параметрів яких лежать за межами значень параметрів навчальних прикладів.

– Можливістю вербалізації навченої НМ, що забезпечує отримання явних правил, за допомогою яких така мережа приймає рішення.

– Можливістю врахування топології піддослідних даних.

– Швидкістю прийняття рішення.

– Необхідним обсягом апаратно-програмних ресурсів.

Базовий варіант переліку критеріїв ефективності, що відповідають зазначеним вимогам, показаний в табл. 1. Запропоновані критерії ефективності мають безрозмірний характер. Надалі зазначений перелік може бути змінений відповідно до конкретних умов задачі розпізнавання вірусів.

За аналогією з [3] прийнято, що значення запропонованих критеріїв можуть змінятися в межах від 0 до 1. При цьому для i-ої архітектури ГНМ значення k-го критерію дорівнює 1, якщо відповідна k-та вимога повністю забезпечується в даній архітектурі, і дорівнює 0, якщо не забезпечується.

Визначення найбільш ефективної архітектури глибокої нейронної мережі. У загальному випадку ГНМ (DNN - Deep Neural Network) - це штучна НМ, у якій кількість схованих шарів більша ніж 2 [1, 2, 4]. Подібно звичайним НМ, ГНМ здатні моделювати складні нелінійні зв'язки між елементами. При цьому, на відміну від звичайних, в процесі навчання ГНМ отримана модель представляє об'єкт у вигляді комбінації простих примітивів. У задачі розпізнавання вірусів подібними примітивами можуть бути виклики потенційно небезпечних API-функцій [4, 5]. Додаткові шари дозволяють моделювати абстракції все більш високих рівнів, що дає можливість будувати моделі для розпізнавання складних об'єктів реального часу.

Таблиця 1

Критерії ефективності виду НММ

Критерій	Вимога
R_1	Можливість використання навчальних прикладів з різною кількістю вхідних параметрів
R_2	Мінімізація обсягу навчальної вибірки
R_3	Можливість використання навчальної вибірки з непропорційним представленням класів, що розпізнаються
R_4	Можливість використання навчальних прикладів без очікуваного вихідного сигналу
R_5	Можливість використання навчальних прикладів, що корелюються
R_6	Пристосованість до донавчання
R_7	Пристосованість до навчання окремими частинами
R_8	Стабільність навчання
R_9	Мінімізація терміну навчання
R_{10}	Забезпечення автоматизації процесу навчання
R_{11}	Здатність навчання на експертних даних
R_{12}	Максимізація обчислювальної потужності
R_{13}	Мінімізація помилки інтерполяції
R_{14}	Мінімізація помилки екстраполяції
R_{15}	Можливість вербалізації результатів
R_{16}	Можливість врахування топології аналізуємих даних
R_{17}	Максимізація швидкості прийняття рішення
R_{18}	Мінімізація необхідного обсягу апаратно-програмних ресурсів

Також однією з відмінностей ГНМ є процес їх навчання. Глибоке навчання - набір алгоритмів, що моделюють високорівневі абстракції в аналізованих даних, використовуючи архітектури, які складаються з множини нелінійних трансформацій.

Слід зазначити, що на сьогодні існує декілька апробованих варіантів архітектур ГНМ. У той же час глибоке навчання швидко розвиваються і нові архітектури, варіанти або алгоритми з'являються досить часто. Однак більшість з них походять від базових архітектур. Тому на першому етапі досліджень доцільно визначити ту архітектуру ГНМ, яка буде найбільш ефективно вирішувати завдання розпізнавання вірусів.

Відповідно до результатів [2, 3], можна виділити три базових архітектури ГНМ:

ГНМ з переднавчанням - A_1. Структурно A_1 повторює багатошаровий персептрон. Використовується функція активації типу гіперболічного тангенсу (15) або сигмоїдальна (16):

$$y(x) = \frac{e^{\alpha x} - 1}{e^{\alpha x} + 1}, \quad (15)$$

$$y(x) = \frac{1}{1 + e^{-\alpha x}}, \quad (16)$$

де $y(x)$ - значення функції активації, x – сумарний вхідний сигнал нейрона, $y(x)$ - деякий коефіцієнт.

Відмінність в архітектурі полягає лише в тому, що процес навчання розділений на два етапи. На першому етапі відбувається попередня настройка вагових коефіцієнтів. Для цього використовуються навчальні приклади, в яких очікуваний вихідний сигнал відсутній. На другому етапі методом «з учителем» реалізується остаточне визначення зазначених вагових коефіцієнтів.

Без переднавчання – A_2. Основною відмінністю такої мережі від багатошарового персептрона є використання так званої випрямленої лінійної функції активації (ReLU)

$$y(x) = \max(0, x). \quad (17)$$

Похідна ReLU дорівнює або 0, або 1, від чого її застосування запобігає розростання і загасання градієнтів, і призводить до проріджування ваг, що позитивно позначається на обчислювальній здатності мережі. Відзначимо, що існує сімейство різних модифікацій ReLU, що вирішують проблеми надійності цієї передавальної функції при проходженні через нейрон великих градієнтів: Leaky ReLU, Parametric ReLU, Randomized ReLU.

Згорткові нейронні мережі - A_3. В цілому така мережа являє собою багатошаровий персептрон, структура якого модифікована (проріджена) з точки зору ієрархічного вилучення ознак.

З урахуванням сформованих базових архітектур ГНМ, критеріїв ефективності (показаних в табл. 1) і виразів (15, 16) для знаходження найбільш ефективної архітектури необхідно для кожного варіанта архітектури ГНМ визначити значення критеріїв ефективності, а також вагові коефіцієнти кожного зі згаданих критеріїв. Для ви-

значення значень критеріїв ефективності використані результати аналізу можливостей нейромережевих архітектур ГНМ, що описані в [8]. Отримані значення критеріїв ефективності для трьох апробованих варіантів архітектур ГНМ показані в табл. 2. Відзначимо, що представлені критерії мають бінарний характер. Критерій дорівнює 0, якщо відповідна вимога не забезпечується нейромережевої архітектурою, і дорівнює 1 в іншому випадку.

Таблиця 2

Значення критеріїв ефективності

Критерій	Значення критерію для виду архітектури		
	Архітектура A_1	Архітектура A_2	Архітектура A_3
R_1	0	0	0
R_2	1	1	1
R_3	1	1	1
R_4	1	0	0
R_5	1	1	1
R_6	1	1	0
R_7	1	1	0
R_8	1	1	1
R_9	1	1	1
R_{10}	1	1	1
R_{11}	0	0	0
R_{12}	1	1	1
R_{13}	1	1	1
R_{14}	1	1	1
R_{15}	0	0	0
R_{16}	0	0	1
R_{17}	1	1	1
R_{18}	1	1	1

Як видно з табл. 2, основні відмінності проаналізованих архітектур полягають у різній пристосованості до вимог, які відповідають критеріям R_4, R_6, R_7, R_{16}. При визначенні вагових коефіцієнтів враховано, що кожен *i*-ий коефіцієнт вказує на значущість *i*-ої вимоги завдання розпізнавання, що відображено відповідним критерієм ефективності.

Визначення найбільш ефективного виду архітектури ГНМ дозволяє перейти до останнього етапу розробки НММ – визначення параметрів цієї архітектури. Зазначимо, що для цього можливо використати результати [5, 11].

Метод розробки архітектури глибокої нейронної мережі, призначеної для розпізнавання вірусів. В результаті проведених досліджень з урахуванням загальновідомої методології створення нейромережевих засобів захисту інформації [5] можна стверджувати, що метод розробки ГНМ призначеної для розпізнавання комп'ютерних вірусів повинен складатись із наступних етапів:

1. Визначення доцільності використання нейромережевої моделі типу ГНМ. Для цього слід використати математичне забезпечення представлене виразами (1-14).

2. Визначення значущості кожного із критеріїв ефективності, представлених в табл. 2.

3. Визначення за допомогою виразів (13, 14) найбільш ефективного виду архітектури нейромережевої моделі типу ГНМ.

4. Визначення параметрів архітектури найбільш ефективного виду.

5. Експериментальне підтвердження достовірності запропонованих рішень.

Вхідними даними методу являються параметри, що характеризуються очікувані умови застосування ГНМ в антивірусних засобах, а виходом – вид та параметри архітектури ГНМ.

Експериментальні дослідження методу розробки архітектури глибокої нейронної мережі.

З урахуванням результатів [9-11] прийнято наступні умови застосування ГНМ:

– НММ використовується для розпізнавання Windows-орієнтованих комп'ютерних вірусів на основі аналізу використаних програмою потенційно небезпечних API-функцій операційної системи;

– на вхід НММ подається інформація, отримана в результаті сканування піддослідних файлів;

– для навчання та тестування НММ використовується опублікована компанією Microsoft база даних комп'ютерних вірусів BIG- 2015;

– допустимий термін створення НММ складає 1 місяць.

Зазначимо, що в БД BIG-2015 представлено приклади сигнатур 9 комп'ютерних вірусів, характеристики яких наведено в табл. 3.

БД BIG-2015 сформована за допомогою програмного комплексу Interactive DisAssembler, що дозволяє вилучити із бінарного файлу метадані які стосуються інструкцій мови Assembler, вміст регістрів та дані і функції, імпортовані із DLL. При цьому застосування до дизасембльованого коду технології Flirt дозволяє визначити наявність в ньому потенційно небезпечних функцій управління розділами, управління файлами, роботи з реєстром, використання системної інформації, використання мережевих з'єднань, управління пам'яттю,

використання сервісів, управління системою захисту об'єктів. В якості прикладу можна навести функцію DeleteFile, що може бути використана для нанесення шкоди файловій системі. Наведений в [5, 7, 9] перелік таких функцій в першому наближенні становить 300 найменувань. Окреслення умов застосування дозволило перейти до реалізації методу розробки ГНМ.

Оскільки для формування навчальної вибірки передбачається використання доступних баз даних, то в виразі (11) величина $\bar{t}_v = 0$. При цьому базуючись на даних [5] визначено, що $\tau = 0{,}01\,\text{с}$. Підставивши ці дані в вираз (11) отримано $t_\Sigma(\mathbf{net}) \approx 8{,}4 \times 10^4$ с. Оскільки $t_d = 2{,}6 \times 10^6$ с, то умова (12) істинна і доцільність використання ГНМ доведена. На наступному етапі розробки було проведене визначення значущості кожного із критеріїв ефективності, представлених в табл. 2. При цьому оцінка значущості кожного з критеріїв була реалізована за допомогою експертного методу парного порівняння. Отримані результати показані в табл. 4.

Таблиця 3

Характеристика БД BIG-2015

Назва вірусу	Кількість навчальних прикладів	Тип вірусу (класифікація компанії Microsoft)
Ramnit	1541	Worm
Lollipop	2478	Adware
Kelihos_ver3	2942	Backdoor
Vundo	475	Trojan
Simda	42	Backdoor
Tracur	751	TrojanDownloader
Kelihos_ver1	398	Backdoor
Obfuscator.ACY	1228	Any kind of obfuscated malware
Gatak	1013	Backdoor

Таблиця 4

Вагові коефіцієнти критеріїв ефективності виду ГНМ в задачі розпізнавання вірусів

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
0,08	0,07	0,02	0,05	0,05	0,14	0,03	0,05	0,05
α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}	α_{16}	α_{17}	α_{18}
0,02	0,05	0,05	0,07	0,07	0,05	0,05	0,05	0,05

З урахуванням даних табл. 2 і табл. 4, функції ефективності апробованих видів ГНМ, розраховані за допомогою виразу (14) дорівнюють $V_{A_1} = 0{,}77$, $V_{A_2} = 0{,}72$, $V_{A_3} = 0{,}6$. Таким чином, при розпізнаванні вірусів найбільш ефективною архітектурою є ГНМ з переднавчанням. Слід зазначити, що найбільшу ефективність даної архітектури можна пояснити можливістю донавчання в процесі експлуатації.

На підставі результатів [8, 11] в основу розробки параметрів архітектури ГНМ покладено тришаровий персептрон, для переднавчання якого використовується розріджений автокодувальник, структура якого показана на рис. 1.

Відзначимо, що на рис. 1 вихідні сигнали вхідних нейронів позначені міткою «x», схованих нейронів - міткою «a», вихідних - «y», а блоків зміщення - міткою «+1». Вхідними даними автокодувальника є нерозмічена навчальна вибірка $x = (x_1, x_2, \ldots, x_i)$. У схованих і вихідних нейронах використовується сигмоїдальна функція активації виду:

$$f(z_k) = \frac{1}{1 + e^{-z_k}}, \qquad (18)$$

де z_k – сумарний вхідний сигнал k-го нейрона в прихованому або вихідному шарі.

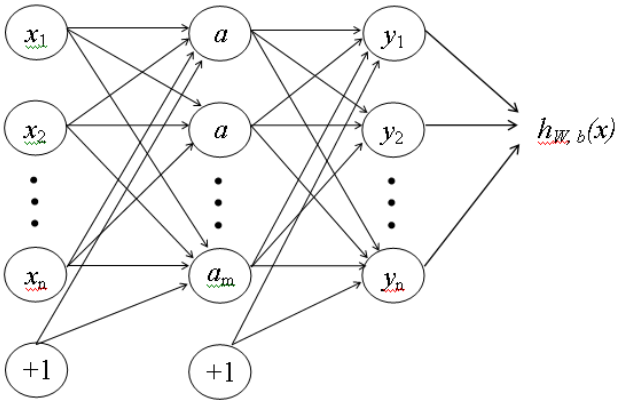

Рис. 1. Архітектура автокодувальника

В свою чергу

$$z_k = \sum_{i=1}^{n} \left(w_{i,k} x_{i,k} + x_0 b_k \right), \qquad (19)$$

де $w_{i,k}$ – вага зв'язку від i-го нейрона попереднього шару до *k*-го нейрона в прихованому або вихідному шарі, $x_{i,k}$ – вхідний сигнал від *i*-го нейрона попереднього шару до *k*-го нейрона, $x_0=1$ – вага зв'язку нейрона з самим собою, b_k – зміщення *k*-го нейрона.

Вихідний сигнал автокодувальника з кількістю нейронних шарів *l* дорівнює

$$h_{W,b}(x) = a^{(l)}, \qquad (20)$$

де *W* – масив вагових коефіцієнтів, b – масив зміщень, $a^{(l)}$ – масив вихідних значень нейронів в шарі *l*.

Особливістю автокодувальника є застосування навчання без вчителя при використанні алгоритму зворотного поширення помилки. Цільова функція навчання автокодувальника визначається виразом:

$$h_{W,b}(x) \approx x. \qquad (21)$$

Використання цільової функції виду (21) передбачає рівність вихідного сигналу автокодувальника вхідного сигналу. Таким чином, навчання класичного автокодувальника зводиться до того, що б за допомогою алгоритму зворотного поширення помилки знайти такі значення вагових коефіцієнтів, при яких вихідний сигнал буде дорівнювати вхідному. При цьому навчальні приклади можуть бути немаркованими, тобто не містити очікуваний вхідний сигнал. Пошук оптимального значення вагових коефіцієнтів проводиться за допомогою градієнтного спуску шляхом мінімізації функції втрат:

$$J(W,b) = \left[\frac{1}{m} \sum_{i=1}^{m} \left(0,5 \left\| h_{W,b}\left(x^{(i)}\right) - y^{(i)} \right\|^2 \right) \right] + 0,5\lambda \sum_{l=1}^{m} \sum_{i=1}^{S_l - 1} \sum_{j=1}^{S_l} \left(w_{j,i}^{(l-1)} \right), \qquad (22)$$

де *m* – кількість схованих шарів, s^l – кількість нейронів в шарі *l*, $w_{j,i}^{(l-1)}$ – вага зв'язку між нейроном *i* в шарі *l* та нейроном *j* в шарі (*l*-1).

Перша частина функціоналу - усереднена квадратична помилка за всіма навчальними прикладами, друга частина регуляризація (або контроль згасання ваг), яка контролює порядок ваг і протидіє процесу перенавчання. Параметр λ, який контролює згасання ваг, регулює відносну важливість двох частин функціоналу.

Навчання проводиться до тих пір, поки:

$$J(W,b) < \theta, \qquad (23)$$

де θ – заздалегідь визначений коефіцієнт (поріг).

Відносно класичного варіанту особливістю розрідженого автокодувальника є обмеження кількості одночасно активних нейронів в проміжних шарах. Вважається, що за рахунок цього розріджений автокодувальник автоматично навчається виділяти з вхідних даних загальні ознаки, які відображаються в значеннях вагових коефіцієнтів. Для цього у функцію втрат вводиться додаткова компонента:

$$P = \sum_{j=1}^{h} \left(p \times \lg \frac{p}{\hat{p}_j} + (1-p) \times \lg \left(\frac{1-p}{1-\hat{p}_j} \right) \right), \qquad (24)$$

де $\hat{p}_j$ – середнє значення функції активації нейрона j з усіх навчальних прикладів, $p \approx 0,05$ – параметр розрідженості.

Відзначимо, що нейрон вважається активним, якщо його вихідний сигнал близький 1, а неактивним - близький до 0. З урахуванням (24) функція втрат розрідженого автокодувальника має вигляд:

$$J_s(W,b) = J(W,b) + \beta P, \qquad (25)$$

де β – заданий коефіцієнт (в першому наближенні $\beta \approx 3$).

Переднавчання ГНМ, у якій кількість нейронних шарів дорівнює *m*, реалізується так:

1. Випадковим чином ініціалізуються вагові коефіцієнти всіх синаптичних зв'язків.
2. Виходячи з необхідної точності навчання встановлюється значення коефіцієнта θ.
3. Встановлюється номер шару, що навчається, $l = 2$ (вхідний шар має номер 1).
4. До *l*-го шару нейронів підключається новий додатковий шар.
5. На вхід *l*-го шару подається множина навчальних прикладів.
6. За допомогою (18-25) розраховуються значення матриці вагових коефіцієнтів зв'язків *l*-го шару нейронів.
7. Підключений на 4 етапі шар нейронів видаляється.
8. Якщо $l < m$, то $l = l + 1$ і здійснюється перехід на 5 етап. В іншому випадку переднавчання закінчується.

Після етапу переднавчання два останніх шари ГНМ навчаються на маркерованих даних.

Значення структурних параметрів побудованої ГНМ розраховані за допомогою даних наведених в [5, 7, 8, 10, 11]. Кількість вхідних параметрів відповідає кількості потенційно небезпечних API-

функцій і $N_x = 300$. Кількість вихідних параметрів відповідає кількості вірусів, що мають бути розпізнані і $N_y = 9$. Кількість схованих нейронних шарів обрано з позицій максимального спрощення структури НМ і дорівнює $K_h = 2$. Кількість нейронів у кожному із схованих шарів розраховується так:

$$N_h = \text{Round}\left(\frac{\sqrt{P \times N_x}}{N_y}\right), \qquad (26)$$

де P – кількість навчальних прикладів.

Відзначимо, що в БД кількість навчальних прикладів, що відповідають вірусам дорівнює 10868. Враховуючи, що в навчальній вибірці необхідне пропорційне представлення зразків комп'ютерних вірусів та безпечних програм визначено, що P=2×10868=21736. Підставивши цю величину в (26), отримано:

$$N_h = \text{Round}\left(\frac{\sqrt{21736 \times 300}}{9}\right) = 284. \qquad (27)$$

Розроблена модель реалізована у вигляді відповідного програмного забезпечення. Програмний код комплексу написаний на мові програмування Python. Вибір мови програмування обумовлений його апробованістю в задачах машинного навчання. Також в процесі розробки програмного забезпечення використана додаткова бібліотека TensorFlow (розробка компанії Google). Ця бібліотека дозволяє автоматизувати більшість операцій, пов'язаних з навчанням і розпізнаванням різних видів нейромережевих моделей. Додатковими перевагами бібліотеки є її безкоштовність та відкритий програмний код. Для експериментів використано персональний комп'ютер (AMD FX-9800P (2.7 - 3.6 ГГц) / RAM 8 ГБ / HDD 1 ТБ / AMD Radeon RX 540, 2 ГБ), що функціонував під управлінням операційної системи Windows 10.

Навчання проводилось на протязі 100 епох. Приблизно після 90 навчальних епох помилка навчання стабілізувалась на рівні 0.01. Після цього на вхід ГНМ із БД BIG-2015 були подані тестові приклади, що не використовувались при навчанні. Похибка розпізнавання для різних вірусів показана на рис. 2.

Аналіз рис. 2 вказує на те, що найбільша похибка розпізнавання характерна для вірусів Simda, Tracur та Vundo. Це можна пояснити невеликою кількістю навчальних прикладів, що відповідають цим вірусам. При цьому середня похибка розпізнавання всіх видів вірусів дорівнює 0,036. Також слід зазначити, що за рахунок використання запропонованого методу при розробці НМ вдалось уникнути довготривалих чисельних експериментів спрямованих на визначення доцільності її використання та на оптимізацію її структурних параметрів. Враховуючи, що досягнута похибка розпізнавання відповідає похибці сучасних антивірусних засобів [5, 9-11], це свідчить про ефективність запропонованих рішень.

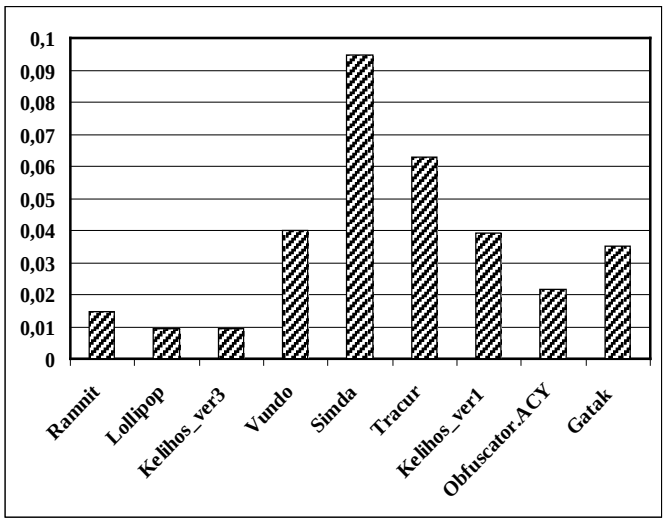

Рис. 2. Похибка розпізнавання на тестовій виборці

Висновки

Показано, що одним з найбільш перспективних напрямків розвитку систем розпізнавання вірусів є удосконалення їх математичного забезпечення за рахунок застосування сучасних нейромережевих моделей на базі глибоких нейронних мереж. Визначено необхідність створення методу розробки такої моделі, адаптованої до умов застосування в антивірусних засобах. Запропоновано метод розробки архітектури глибокої нейронної мережі, призначеної для розпізнавання вірусів. На відміну від існуючих метод дозволяє уникнути в процесі розробки нейромережевої моделі довготривалих чисельних експериментів спрямованих на визначення доцільності її застосування та на оптимізацію її структурних параметрів. При цьому шляхом чисельних експериментів з використанням опублікованої компанією Microsoft бази даних комп'ютерних вірусів BIG-2015 показано, що метод дозволяє побудувати нейромережеву модель, яка забезпечує похибку розпізнавання, співрозмірну з похибкою сучасних систем розпізнавання комп'ютерних вірусів. Перспективи подальших досліджень пов'язані з адаптацією запропонованого методу до застосування глибоких нейронних мереж в поведінкових аналізаторах.

ЛІТЕРАТУРА

[1]. А. Артеменко, В. Головко, "Анализ нейросетевых методов распознавания компьютерных вирусов", *Молодежный инновационный форум «ПИНТРИ»*, Минск, ГУ «БелИСА», 2010, 239 с.

[2]. М. Баклановский А. Ханов, К. Комаров, П. Лозов, "Оценка точности алгоритма распознавания вредоносных программ на основе поиска аномалий в работе процес сов", *Научно-технический вестник информационных технологий, механики и оптики*, Т. 16., № 5, С. 823-830, 2016.

[3]. В. Вишняков, О. Коваль, М. Моздурани, "Использование нейронных сетей для обнаружения и распознавания аномалий в корпоративной информационной системе предприятия", *Доклады Белорусского государственного университета информатики и радиоэлектроники,* № 4 (98), С. 152-160, 2016.

[4]. А. Киселевская, "Глубокие нейронные сети: автоматическое обучение распознаванию вредоносных программам. Генерация и классификация подписей", *Молодой учёный,* № 47 (181), С. 15-17, 2017.

[5]. А. Корченко, И. Терейковский, Н. Карпинский, С. Тынымбаев, *Нейросетевые модели, методы и средства оценки параметров безопасности интернет-ориентированных информационных систем,* [Монография], Киев, 2016, 275 с.

[6]. С. Поликарпов, В. Дергачёв, К. Румянцев, Д. Голубчиков, "Новая модель искусственного нейрона: кибернейрон и области его применения", *Известия ЮФУ. Технические науки,* № 9 (134), 2012, С. 94-98.

[7]. І. Терейковський, "Нейромережевий поведінковий аналізатор антивірусної системи", *Захист інформації*, № 2, С. 67-70, 2012.

[8]. I. Bapiyev, B. Aitchanov, I. Tereikovskyi, L. Tereikovska, A. Korchenko, "Deep neural networks in cyber attack detection systems", *International Journal of Civil Engineering and Technology (IJCIET)*, Volume 8, Issue 11, November 2017, pp. 1086–1092, 2017.

[9]. M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification", *In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, CODASPY'16*, pp. 183–194, 2016.

[10]. F. Asiru Omotayo, Moses T. Dlamini and Jonathan M. Blackledge Asiru, "Application of Artificial Intelligence for Detecting Derived Viruses", *16th European Conference on Cyber Warfare and Security (ECCWS 2017)*, University College Dublin, Dublin June 29-30, pp. 217-227, 2017.

[11]. Himali Jani, Sathvik Shetty, Kiran Bhowmick, "Virus Detection using Artificial Neural Networks", *International Journal of Computer Applications*, Volume 84, No 5., December 2013, pp. 17-23, 2017.

МЕТОД РАЗРАБОТКИ АРХИТЕКТУРЫ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ, ПРЕДНАЗНАЧЕННОЙ ДЛЯ РАСПОЗНАВАНИЯ КОМПЬЮТЕРНЫХ ВИРУСОВ

Статья посвящена решению задачи совершенствования систем распознавания компьютерных вирусов. Хотя используются системы антивирусной защиты уже не одно десятилетие, их разработкой занимается много высококвалифицированных специалистов, а созданию соответствующей научно-методической базы посвящено большое количество работ, однако практический опыт и известные случаи успешных вирусных атак на отечественные и зарубежные компьютерные системы, и сети указывают на наличие в современных антивирусах распознавания серьезных недостатков. Показано, что исправить ряд недостатков возможно путем совершенствования математического обеспечения процедуры распознавания за счет применения современных нейросетевых моделей на базе глубоких нейронных сетей. Предложен метод разработки архитектуры глубокой нейронной сети, предназначенной для распознавания вирусов. В отличие от существующих метод позволяет избежать в процессе разработки нейросетевой модели длительных многочисленных экспериментов, направленных на определение целесообразности ее применения и на оптимизацию ее структурных параметров. Путем многочисленных экспериментов с использованием опубликованной компанией Microsoft базы данных компьютерных вирусов BIG-2015 показано, что метод позволяет построить нейросетевой модели, обеспечивающей погрешность распознавания, соразмерный с погрешностью современных систем распознавания компьютерных вирусов. Определено, что перспективы дальнейших исследований связаны с адаптацией предложенного метода к применению глубоких нейронных сетей в поведенческих анализаторах

Ключевые слова: защита информации, компьютерный вирус, нейросетевая модель, глубокая нейронная сеть, разреженный автокодировщик.

METHODS FOR DEVELOPING A DEEP NEURAL NETWORK ARCHITECTURE DESIGNED TO RECOGNIZE COMPUTER VIRUSES

The article is devoted to the solution of the problem of improving computer virus recognition systems. Although the antivirus protection systems have been used for several decades, a lot of highly skilled specialists are involved in their development, and a large number of works are devoted to the creation of the appropriate scientific and methodological base, but practical experience and known cases of successful virus attacks on domestic and foreign computer systems and networks point to the presence in

modern antivirus detection of serious shortcomings. It is shown that correcting a number of disadvantages is possible by improving the mathematical support of the recognition procedure due to the use of modern neural network models based on deep neural networks. The method of development of the architecture of the deep neural network intended for the recognition of viruses is proposed. In contrast to the existing method, it is possible to avoid during the development of a neural network model of long-term numerical experiments aimed at determining the appropriateness of its application and optimizing its structural parameters. By numerical experiments using Microsoft's computer virus database BIG-2015 published by Microsoft, it is shown that the method allows constructing a neural network model that provides a recognition error that is commensurate with the error of modern computer virus detection systems. It is determined that the prospects for further research are related to the adaptation of the proposed method to the application of deep neural networks in behavioral analyzers.

Keywords: information security, computer virus, neural network model, deep neural network, rarefied autocoder.

Терейковський Ігор Анатолійович, доктор технічних наук, доцент, професор кафедри системного програмування та спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».
E-mail: terejkowski@ukr.net.
Терейковский Игорь Анатольевич, доктор технических наук, доцент, профессор кафедры системного программирования и специализированніх компьютерніх систем Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского».
Tereikovskyi Ihor, Doctor of Technical Sciences, Associate Professor, Professor of the Department of System Programming and Specialized Computer Systems of the National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorsky".

Заріцький Олег Володимирович, доктор технічних наук, доцент кафедри безпеки інформаційних технологій Національного авіаційного університету.
E-mail: olegzaritskyi@gmail.com.
Зарицкий Олег Владимирович, доктор технических наук, доцент кафедры безопасности информационных технологий Национального авиационного университета.
Zaritskyi Oleg, Doctor of Engineering Sciences, Associate Professor of Academic Department of IT-Security, National Aviation University (Kyiv, Ukraine).

Терейковська Людмила Олексіївна, кандидат технічних наук, доцент, доцент кафедри кібернетичної безпеки та комп'ютерної інженерії Київського національного університету будівництва і архітектури.
E-mail: tereikovskal@ukr.net.
Терейковская Людмила Алексеевна, кандидат технических наук, доцент, доцент кафедры кибернетической безопасности и компьютерной инженерии Киевского национального университета строительства и архитектуры.
Tereikovska Liudmyla, Candidate of Technical Sciences, Associate Professor, Associate Professor of Cyber Security and Computer Engineering at the Kyiv National University of Construction Architecture.

Погорелов Володимир Володимирович, асистент кафедри безпеки інформаційних технологій Національного авіаційного університету.
E-mail: volodymyr.pogorelov@gmail.com.
Погорелов Владимир Владимирович, ассистент кафедрs безопасности информационных технологий Национального авиационного университета.
Pogorelov Vladimir, Assistant of Information Security Departments of the National Aviation University.